

Microcontroller-Based Serial Port Interface (SPI®) Boot Circuit

by Alfredo Barriga

INTRODUCTION

This application note describes the operation of a general-purpose, microcontroller-based Serial Port Interface (SPI) boot circuit. This is a low cost solution for users who need to modify some of their device's parameters at power up. This circuit addresses a 3-wire SPI application for programming converters, or any device that has a SPI option, and sends commands to user-defined SPI registers.

SPI DEFINITIONS

The SPI port consists of three pins. These are the serial clock pin (SCLK), the serial data IO pin (SDIO) and the chip select bar pin (CSB). SCLK is the serial shift clock in and is used to synchronize serial interface reads and writes. SDIO is a dual-purpose pin, which is used as an input and output depending on the instruction being sent and the relative position in the timing frame. CSB is an active low control that gates the read and write cycles. Figure 1 shows the SPI timing details. Table 1 provides specifications.

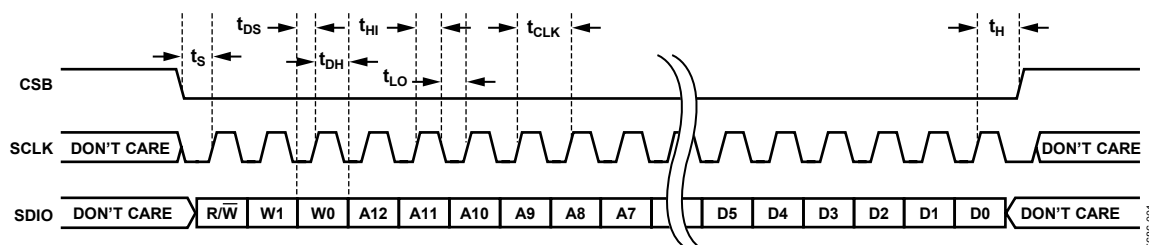


Figure 1. SPI Timing Details

Table 1.

Specification Name	Meaning
t_{DS}	Setup time between data and rising edge of SCLK
t_{DH}	Hold time between data and rising edge of SCLK
t_{CLK}	Period of the clock
t_S	Setup time between CSB and SCLK
t_H	Hold time between CSB and SCLK
t_{HI}	Minimum period that SCLK should be in a logic high state
t_{LO}	Minimum period that SCLK should be in a logic low state

TABLE OF CONTENTS

Introduction 1

SPI Definitions 1

 Revision History 2

Circuit Description..... 3

 Configuration File 3

 Circuit Operation 4

Programming the Device..... 5

 Off-PCB Programming..... 5

 Voltage Level Translation 6

 Obtaining Code and Programming Tools..... 6

Component Details and BOM 7

References 8

REVISION HISTORY

12/05—Revision 0: Initial Version

CIRCUIT DESCRIPTION

The core of this particular SPI boot circuit is the PIC12F629 microcontroller. This Flash™-based device is capable of writing to the SPI port using the SCLK, SDIO, and CSB control signals. It can be programmed many times to fit different application needs, and because of its small size, can be easily incorporated on any design.

CONFIGURATION FILE

The configuration file to be programmed into the chip is automatically generated by a standalone LabVIEW™ application which allows the user to specify the number of registers to be accessed and the 8-bit data-word to be written on each location. LabView generates the assembly code which is then compiled and linked to produce the hex file to be programmed into the microcontroller.

There are two applications used for this purpose, which are called by the main LabVIEW standalone configuration program (SPI Code Builder.exe). MPASM® by Microchip® is used to compile and link the assembly program, and PicFLASH® by MikroElektronika® serves as a programmer to load the generated hex file into the microcontroller. These two Windows®-based applications have been included in the SPI Code Builder application setup file. The user may also employ any other compiler and programming application to assemble the source code provided by the standalone program. Figure 2 shows an example of the SPI Code Builder and PicFLASH® Programmer windows.

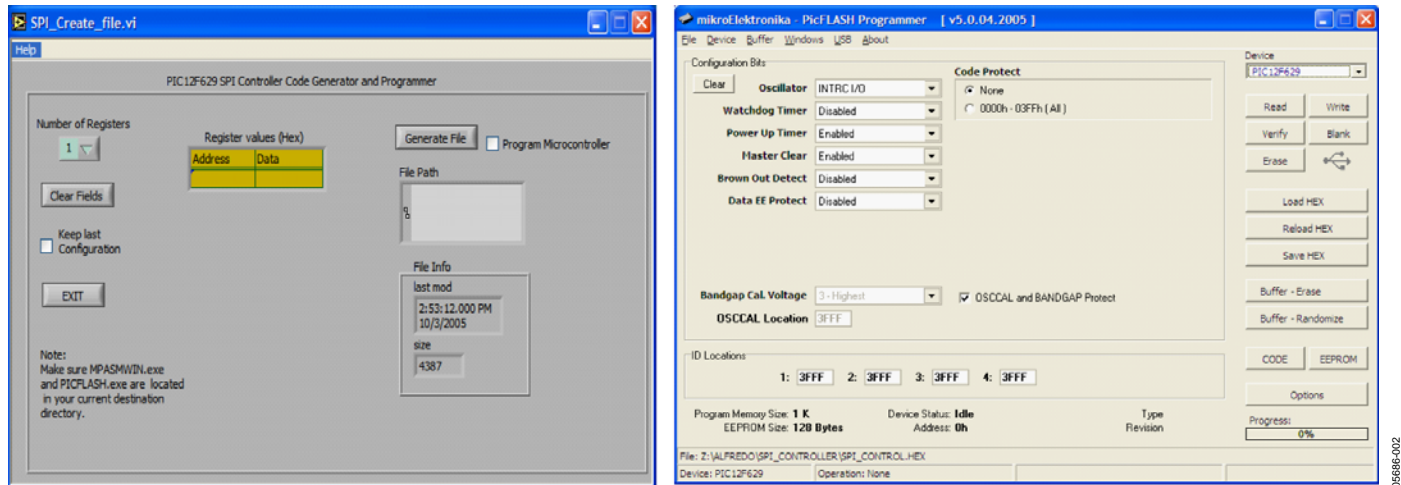


Figure 2. SPI Code Builder and PicFLASH® Windows

PROGRAMMING THE DEVICE

The microcontroller used in this implementation can be serially programmed while in the end application circuit. This is done with two lines for clock and data, and three lines for power, ground, and programming voltage/RESET. There are a number of low cost serial and USB programmers available in the market, such as the PicFLASH®USB programmer by MikroElektronika®. This programmer can be purchased online at a low cost.

Programming specifications require that the unit be powered at 5 V during programming. If the microcontroller is operated at a voltage smaller than 5 V, a switch or a two-position jumper must be added to select from system supply and programming supply (SW1). A three-position switch allows the circuit to be in three different modes—programming mode, normal SPI mode, and power off. The off position can be incorporated if the user does not want to write to the SPI port when the system is powered up. To eliminate the need of switching between programming supply and system supply, the microcontroller can be permanently powered at 5 V. This adds simplicity to the design and at the same time complies with the programming voltage requirement.

OFF-PCB PROGRAMMING

The user may choose to program the microcontroller before soldering it to the board. This can be done by using the appropriate socket for the package selected (8-lead PDIP, SOIC, or DFN-S). The control lines used by the programmer can be directly connected into the socket for off-PCB programming. Figure 3 shows a simplified schematic using a preprogrammed microcontroller. In this circuit, VDD is directly connected to the 3.3 V supply and the RESET button is permanently removed from MCLR (Pin 4). The user can choose to enable or disable the microcontroller by adding a jumper between the supply and the VDD pin.

Another advantage of programming the microcontroller off-PCB is that there is no worry about setting the microcontroller to a voltage that matches the required programming supply voltage. The device can be powered at any voltage between 2 V and 5.5 V, providing the flexibility of using an existing power supply.

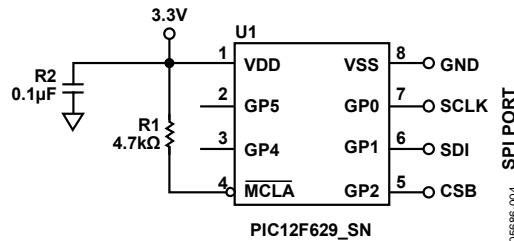


Figure 4. SPI Boot Circuit (Simplified)

VOLTAGE LEVEL TRANSLATION

Logic thresholds are dictated by the process and design implemented on each product. On some products, SPI inputs are 5 V or 3.3 V tolerant. However, other products may only accept smaller voltages (for example, 1.8 V). In this case, it might be necessary to incorporate a voltage level translator to adapt the levels from the microcontroller to levels acceptable to the inputs of the DUT. It is recommended that the customer uses the ADG3304 which is a four-channel bidirectional level translator. The translator can be customized to perform bidirectional logic level translation without an additional signal to set the direction in which the translation takes place. For instance, if the microcontroller is operating at 5 V supply and

the SPI port is 1.8 V tolerant, the ADG3304 can be set so that it translates the SCLK, SDI, and CSB signals from 5 V to 1.8 V. The ADG3304 is an easy-to-use solution that requires very few external components. Moreover, the user can disable the outputs from the translator at any time by pulling Pin 8 (EN) low. Figure 5 shows the implementation of the SPI boot circuit with the voltage level translator.

OBTAINING CODE AND PROGRAMMING TOOLS

Source code and installer with programming tools can be found at no cost at www.analog.com. For information about the PicFLASH USB programmer, go to www.mikroelektronika.co.yu.

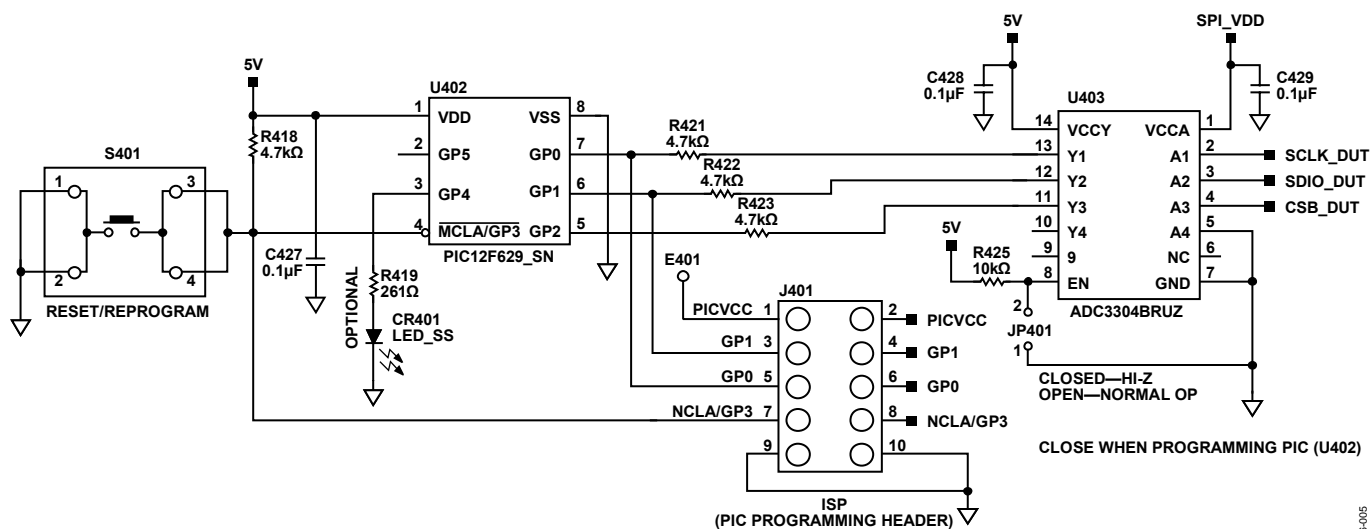


Figure 5. SPI Boot Circuit with Translator

COMPONENT DETAILS AND BOM

Table 2. Recurring Components

QTY	DEVICE
4	RES_402
1	SWITCH01 ¹
1	PIC12F629_SOIC
1	HEADER10

¹ Any three-position switch. May also use a jumper block.

Table 3. One Time Purchases

QTY	DEVICE
1	PIC Flash USB Programmer ¹

¹ Or any PIC compatible serial programmer.

Table 4. Optional Components

QTY	DEVICE
1	8-Lead SOIC Socket
1	LED_GAAS
1	RES_402
1	ADG3304
1	PUSHBUTTON

REFERENCES

- *Interfacing to High Speed ADCs via SPI® User Manual* is available at www.analog.com.
- The ADG3304 data sheet is available at www.analog.com.
- The PIC12F629 data sheet and *MPLAB® PM3 ICSP Design Guide* can be obtained from Microchip® at www.microchip.com.
- Information on the PicFLASH® USB Programmer and open source software can be obtained at www.mikroelektronika.co.yu